Amendments to the Specification:

Please replace the current Brief Description of the Drawings section (starting on page 8) with the following replacement section:

BRIEF DESCRIPTION OF THE DRAWINGS

The invention may be understood by reference to the following description taken in conjunction with the accompanying drawings, in which reference numerals identify like elements, and in which:

Figure 1 depicts one particular embodiment of a Future I/O SAN constructed, configured, and operated in accordance with the present invention;

Figures 2-3 illustrate the data transfer concepts of "channel semantics" and "memory semantics" in the context of the present invention;

Figure 4 FIO Client Processes Communicates With FIO Hardware Through Queue Pairs 28;

Figure 5 Connected Queue Pairs;

Figure 6 Connectionless Queue Pairs;

Figure 7 Multiple SANICs per host and multiple ports per SANIC;

Figure 8 Identifying Names for LLEs, SANICs, etc.;

Figure 9 Subnets and Local Identifiers ("LIDs");

Figure 10 Paths Within and Among Subnets;

Figure 11 A FIO message partitioned into Frames and Flits;

Figure 12 Multiple Request Frames (and Flits) and Their Acknowledgment Frames (and Flits);

Figure 13 Single Board Computer;

Figure 14 Remote I/O – Active Backplane;

Figure 15 Remote I/O – Passive Backplane;

Figure 16 Chassis-to-Chassis Topology;

Figure 17 FIO Architecture Layers;

Figure 18 FIO SAN, Software and Hardware Layers;

Figure 19 Future I/O Layered Architecture;

Figure 20 Data flow of flit delimiters and flit body data in flit layer;

Figure 21 Flit Delimiter Fields;

Figure 22 Flit TYPE definition;

Figure 23 Flit Transmission Priority;

Figure 24 FIO Data Path and Interfaces for Link and Physical Layers;

Figure 25 8b/10b Coding Conversion;

Figure 26 Beacon Sequence;

Figure 27 FIO Link Training – One End Power-on;

Figure 28 Future I/O Layered Architecture;

Figure 29 Sample Point-to-point Topologies;

Figure 30 Single-board Platform;

Figure 31 Passive Backplane Platform;

Figure 32 Platform-to-Chassis Topology;

Figure 33 endnodes Connected via External Switch Elements;

Figure 34 Leaf End-station with Embedded Switch;

Figure 35 Sample Switch Frame Routing;

Figure 36 Sample Router Route Operation;

Figure 37 Attribute Comparison of Switch Routing Technologies;

Figure 38 Graphical View of Route Headers and Payloads;

Figure 39 Sample Unreliable Multicast Send Operation;

Figure 40 Policy Based Routing Example;

Figure 41 Connected QPs for Reliable Transfer Operation;

Figure 42 Connectionless QPs for Reliable Datagram Operation;

Figure 43 Connectionless QPs for UnReliable Datagram Operation;

Figure 44 QP State Diagram;

Figure 45 Completion queue model;

Figure 46 Connection establishment accept frame transmission sequence;

Figure 47 Connection establishment reject frame transmission sequence;

Figure 48 Connection teardown frame transmission sequence;

Figure 49 QP Attribute Update frame transmission sequence;

Figure 50 – Port States in Spanning Tree;

Figure 51 – Path Costs to Root (seen by Switch E);

Figure 52 – Bundles;

Figure 53 Node Configuration and Discovery;

Figure 54 Boot Process;

Figure 55 Diagram highlighting communication between Application, Agent and FIO devices;

Figure 56 A block diagram of a managed server;

Figure 57 Block diagram of a FIO management console and FIO management server;

Figure 58 Block diagram of Parition-1;

Figure 59 Diagram of communication flow between Partition-1 and Partition-2;

Figure 60 FIO management routing diagram;

Figure 61 IPv6 interoperability diagram;

Figure 62 Example Endpoint Partitions of an FIO-Connected Cluster;

Figure 63 Simple Tree with Mixed Bandwidth Links and Adapter Leaves;

Figure 64 Simple Tree with Mixed Bandwidth Links and Adapter and Router Leaves;

Figure 65 – Simple Tree with Mixed Bandwidth Links and Adapter and Router Leaves;

Figure 66 illustrates one particular embodiment of a method for training a link for communicating information in a computing system;

Figure 67 illustrates a method for training communications links in one aspect of the method in Figure 66;

Figure 68 depicts an exemplary embodiment of a computing system on which the method of Figure 66 may be implemented;

Figure 69 shows a functional block diagram of the computing system in Figure 68;

Figure 70 shows a functional block diagram of computing system alternative to that in Figure 69;

Figure 71 shows a ladder diagram of a training sequence in accordance with the present invention used to train ports;

Figure 72 shows a table of the training sequences that comprise the training sequences used in Figure 71 to train the ports;

Figure 73 shows a table of lane identifiers used to label the individual channels in a serial physical link;

Figure 74 shows a functional block diagram of a serial physical link;

Figure 75 shows a functional block diagram of an adapter configured to transmit and receive differential signals; and

Figure 76 symbolically represents a de-skew method used in the illustrated embodiment.

On page 15, prior to the last paragraph, please insert the following new paragraphs:

The Queue tail, consists of a Send Queue and a Receive Queue, message requests are initiated by posting Work Queue (WQE) to the Send Queue. The FIO client's message is referenced by a gather-list in the Send WQE as shown in Figure 11. Each entry in the gather-list points to a virtually contiguous buffer in the clients local memory space.

Hardware reads the WQE and packetizes the messages into frames and flits. Frames are routed through the network, and for reliable transport services, are acknowledged by the final destination. If not successfully acknowledged, the frame is retransmitted by the source endnode. Frames are generated and consumed by the source and destination endnodes, not by the switches and routers along the way.

Flits are the smallest unit of flow control on the network and are generated and consumed at each end of a physical link. Flits are acknowledged at the receiving end of each link and are retransmitted if there is an error.

The send request message is sent as two frames. Each request frame is in turn broken down into 4 flits. The ladder diagrams shown on Figure 12 show the request and ACK frames going between the source and destination endnodes as well as the requests and ACK flits between the source and destination of each link.

In Figure 12, a message is shown being sent with a reliable transport. Each request frame is individually acknowledged by the destination endnode. The second part of Figure 12 shows the flits associated with the request and acknowledgment frames passing among the processor endnodes and the two FIO switches. An ACK frame fits inside one flit and one acknowledgement flit acknowledges several flits.